28
150 pts.

RUN

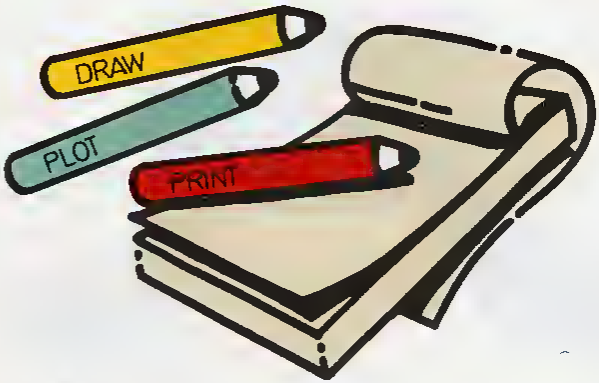
Enciclopedia Práctica del Spectrum

Nueva Lente/Ingelek

INVESTIGANDO LA PANTALLA

HASTA ahora hemos tratado de forma poco profunda la estructura de la pantalla. Sabemos, eso sí, que podemos escribir sobre ella de dos formas, en modo texto por medio de la sentencia **PRINT** con sus calificativos **TAB, AT,** coma (,), punto y coma (;) y apóstrofe ('), o en modo alta resolución a través de las sentencias **PLOT** y **DRAW.**

Con mayor detalle, diremos que la pantella que nosotros vemos consta de dos partes fundamentales: la zona de configuración de puntos y la zona de atributos.



La escritura en la pantalla es posible mediante tres sentencias: **PRINT, PLOT** *y* **DRAW.**

UN SENCILLO EXPERIMENTO

De todo ello nos damos buena cuenta a la hora de cargar en memoria el contenido de una pantalla almacenado en cinta por medio de **SCREEN$.** En el caso de no disponer de ninguna grabada por nosotros, nos puede servir para este experimento cualquiera de presentación de un juego comercial.

En primer lugar, debemos localizar el comienzo de una pantalla en la cinta, cosa que no nos será difícil dado que su sonido es bastante peculiar (más agudo de lo habitual en programas).

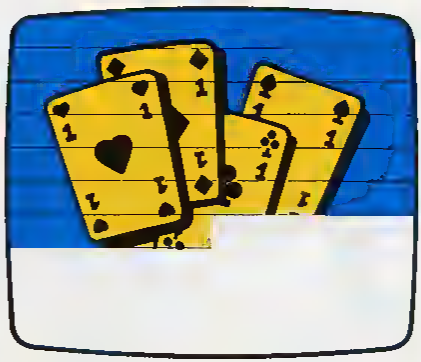
Acto seguido, teclearemos **LOAD " " SCREEN$** seguido de **ENTER,** tras lo cual el ordenador queda preparado para cargar la pantalla desde el casete.

Por último, colocaremos la grabadora en la posición *PLAY,* después de efectuar las oportunas conexiones de *EAR,* y situando el volumen a un nivel adecuado.

Fijándonos detenidamente, podemos observar que el contenido de la cinta, se va representando en la pantalla del televisor por líneas completas, es decir, el primer *byte* recibido se corresponde con la configuración de puntos de la línea superior, de la primera posición de carácter de la pantalla (0,0), y el segundo, a la misma primera fila del segundo carácter de la pantalla (0,1), y así hasta completarse la primera hilera de puntos, al leerse el *byte* número 32.

Siguiendo esta norma, parece que lo más lógico sería que los *BYTES* sucesivos correspondieran a las segundas filas de puntos de los caracteres de

El contenido de la cinta se va representando en la pantalla del televisor o monitor por líneas completas



¡!

La pantalla que nosotros vemos consta de dos partes fundamentales: la zona de configuración de puntos y la zona de atributos.

*

La zona de atributos de 768 *bytes* (24 × 32), almacena los códigos y parpadeo, correspondientes a una posición de carácter determinada.

La localización de una pantalla grabada en una cinta no es difícil, dado que su sonido es más agudo de lo habitual.

la primera línea de la pantalla, pero esto no es así.

En realidad, los próximos 32 *bytes* se corresponden, guardando el mismo orden que en el caso anterior, a la primera fila de la configuración de puntos de la segunda línea de caracteres.

Lo más sorprendente, sin embargo, es que esta situación no se mantiene hasta completarse la totalidad de las primeras filas, de las 24 líneas de caracteres de que se compone la pantalla, sino que termina inmediatamente después de la lectura de la primera fila de puntos de la configuración de los caracteres de la octava línea.

Acto seguido, los *bytes* que llegan van situándose, ordenadamente, a partir de la segunda fila de la configuración de puntos de los caracteres de la primera línea, después de la segunda, y así sucesivamente hasta alcanzar el final de la segunda línea de configuración de los caracteres, de la línea octava de la pantalla.

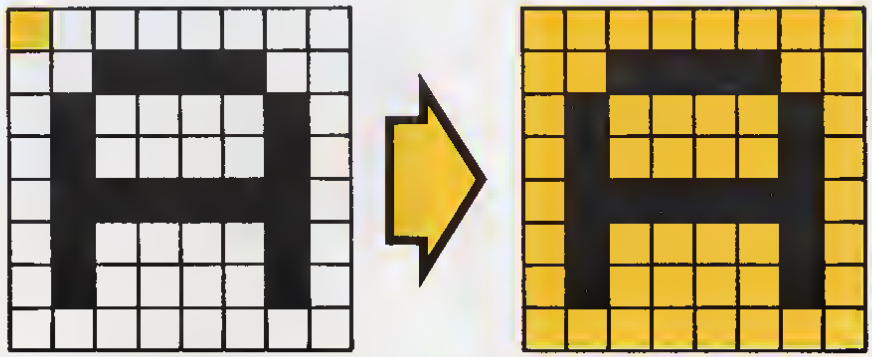
Con cierta monotonía, esta situación se repite hasta completar la totalidad de las ocho líneas de configuración de puntos, de las ocho primeras líneas de impresión, es decir, hasta completarse la lectura de las primeras 2 K de la cinta ($8 \times 8 \times 32 = 2048 = 2K$).

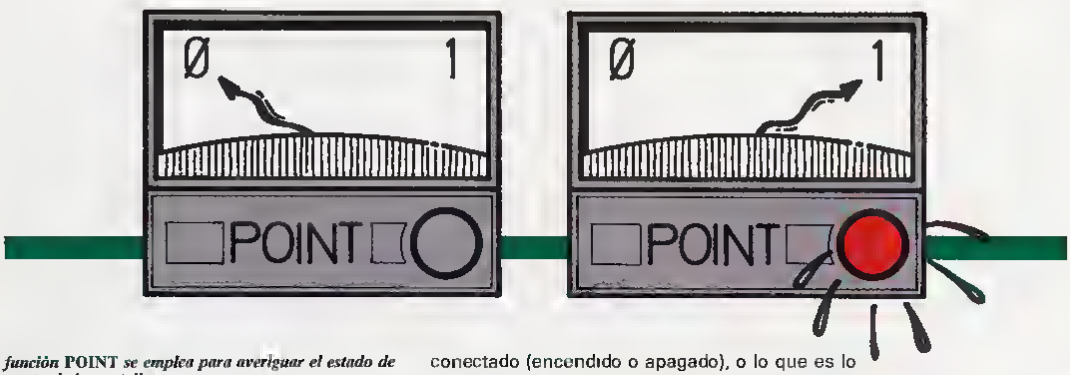
Llegados a este punto, hemos completado con éxito el primer tercio de la pantalla, al menos en lo que se refiere a la configuración de puntos, repitiéndose la misma regla de formación hasta ahora descrita, para el segundo y tercer tercio de pantalla, formados por las líneas de impresión novena a décimo sexta y décimo séptima a vigésimo cuarta respectivamente con lo cual se completan las 6 K de que se compone la configuración total de puntos de la pantalla.

Como final del experimento observaremos que, acto seguido, se leen ordenadamente los 768 *bytes* que contienen los valores de los «atributos» **INK** y **PAPER**, con sus correspondientes calificativos de **BRIGHT** y **FLASH**.

Estos atributos son aplicables a nivel de porción de carácter, debido a lo cual su lectura se realiza de la cinta en el mismo orden en que se produce la impresión normal de caracteres, es decir, de izquierda a derecha y de arriba abajo, carácter a carácter hasta completar la totalidad de la pantalla.

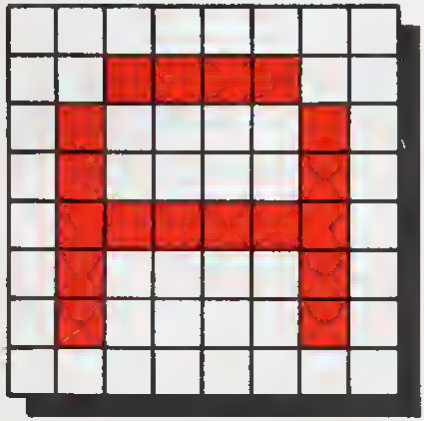
No podemos asignar un color determinado a un pixel de la pantalla, sin que con ello se vea alterado el color de los pixels pertenecientes al mismo carácter.





La función **POINT** *se emplea para averiguar el estado de un punto de la pantalla*

Cada carácter se representa en la pantalla por ocho bytes consecutivos en vertical.



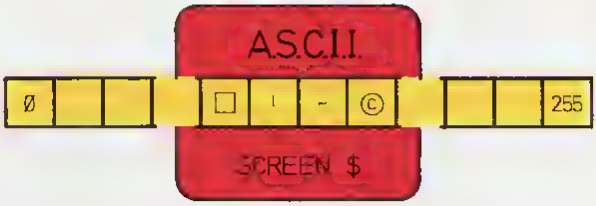
ZONA DE CONFIGURACIDN DE PANTALLA

La zona de configuración está compuesta por una secuencia de 49 152 puntos *(bits),* que corresponden a cada uno de los *pixel* de la pantalla, es decir, al producto de 192 × 256 (24 filas de 32 columnas con caracteres de 8 × 8 *pixels)*
Es evidente que esta información proveniente de la pantalla es claramente binaria, al bastar con conocer de cada punto si está conectado o desconectado (encendido o apagado), o lo que es lo mismo, si adopta el color de la **INK** o del **PAPER** de la zona de carácter que le corresponde.
En total, se reservan 6 144 (24 × 32 × 8) *bytes,* para almacenar la configuración de puntos de la pantalla, en base a que cada una de las posiciones de carácter necesita de 8 bytes (64 *bits)* para su representación
Este espacio de memoria nos va a permitir representar cualquier cosa en la pantalla, tanto en modo carácter como en alta resolución, si no tenemos en cuenta el color, ni el resto de las «propiedades» aplicables a cada zona de carácter, como son **FLASH,** y **BRIGHT.**

ZONA DE ATRIBUTOS DE PANTALLA

Todas estas propiedades son aplicables a nivel de zona de carácter y no de punto, como hemos vis-

SCREEN$ *sólo será aplicable en el entorno del juego de caracteres* **A S.C.I.I.** *convencional*



¡!

La zona de configuración está compuesta por una secuencia de 49 152 puntos *(bits),* que corresponden a cada uno de los *pixel* de la pantalla es decir, al producto de 192 × 256 (24 filas de 32 columnas con caracteres de 8 × 8 *pixels)*

La pantalla se compone de un total de 49 152 elementos de imagen (pixels).

to en páginas anteriores. Esto viene a decir que no podemos asignar un color determinado a un pixel de la pantalla, sin que por ello se vea alterado el color de los *pixel* «vecinos», y lo mismo sucede con el resto de las propiedades aplicables sólo a este nivel.

De esta forma, nos vemos ante una zona de atributos de 768 *bytes* (24 × 32), almacenándose en cada uno de éstos los atributos correspondientes a una posición de carácter determinada.

Una de las razones de que esto suceda de este modo es, sin lugar a dudas, la ocupación enorme que supondría mantener una «memoria de atributos» para cada *pixel* concreto.

Para hacernos una idea, basta con que consideremos que sería necesario un *byte* extra por cada pixel de pantalla, es decir, 393.216 *bytes* (49.152 × 8), o sea 384 K, para que pudiéramos disfrutar de una pantalla en alta resolución en la que las características de color, brillo y parpadeo fueran aplicables, de forma individual, a cada pixel concreto.

¡!

La información referente a la pantalla es claramente binaria, al bastar con conocer de cada punto si está conectado o desconectado (encendido o apagado), o lo que es lo mismo, si adopta el color de la **INK** o del **PAPER** de la zona de caracter que le corresponde.

LECTURA DE LA PANTALLA

Ya conocemos, por capítulos anteriores, las dos formas de que dispone el Spectrum para colocar datos en pantalla. La primera de ellas, en modo carácter, a través de la sentencia **PRINT**. La segunda, en modo alta resolución, por medio de **PLOT**.

Es lógico que, del mismo modo que podemos colocar caracteres y puntos en lugares concretos de la pantalla, podamos examinar también el contenido de estas posiciones a través de otras sentencias.

Esto lo podemos hacer, en la alta resolución, a través de la función **POINT**, seguida como argumento de las coordenadas de un punto determinado, que nos brinda como respuesta si el punto se encuentra encendido o apagado (del color de la **INK** o del **PAPER**). En el primer caso la respuesta será 1, en el segundo, 0.

Del mismo modo que podemos investigar si un punto concreto de la pantalla está encendido o apagado, podemos hacer lo propio con la posición de carácter. En este caso, diremos que se trata de la función inversa a **PRINT**, igual que **PLOT** y **POINT** son funciones inversas.

La sentencia BASIC que nos permite esta averiguación es **SCREEN$**. De forma similar a la función **POINT**, **SCREEN$** debe ir seguida de un par de parámetros separados por una coma y encerrados entre paréntesis. El primero de ellos indica la línea y el segundo la columna de pantalla, de forma análoga al empleo de **AT**.

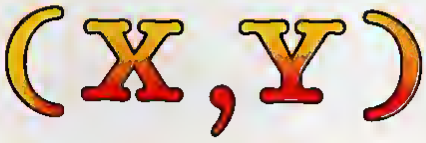
La sentencia nos da como respuesta el carácter que ocupa la posición especificada en el argumento. Por lo tanto, el resultado es una cadena de un carácter, el cual ocupa la posición concreta. El formato general de la sentencia es:

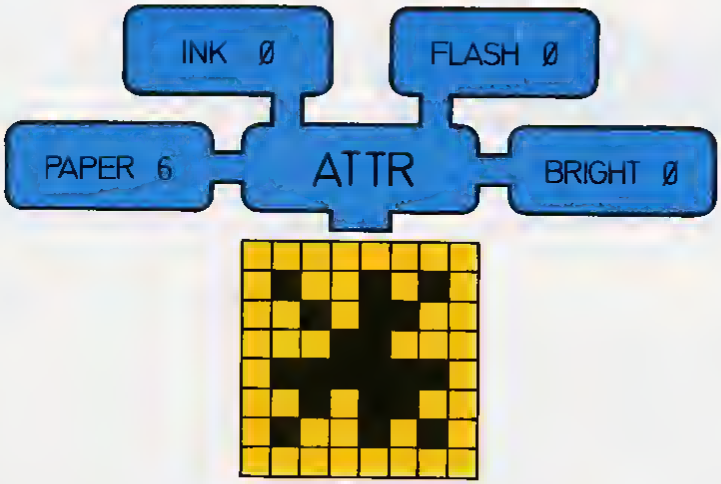
SCREEN$ (X,Y)

Donde X e Y son las coordenadas de la posición de pantalla, y **SCREEN$** puede ir precedido por la sentencia **PRINT**, una asignación a variable por medio de **LET**, alguna sentencia de comparación, etc., según nuestras necesidades.

Sin embargo, **SCREEN$** tiene limitaciones, y cuando especificamos las coordenadas de una posición ocupada por un carácter gráfico, tanto de los incluidos en el teclado como de los definidos por el usuario, obtendremos como respuesta una cadena nula. Por ello, **SCREEN$** sólo será aplicable en el entorno del juego de caracteres ASCII convencional.

Las funciones **ATTR**, **POINT** *y* **SCREEN$** *deben ir seguidas de dos parámetros separados por una coma y encerrados entre parentesis*





ATTR se emplea para investigar la configuración de los atributos en la pantalla.

Pero además de la posibilidad de investigar la zona de configuración de puntos de la pantalla, podemos entrar también en la de atributos, a través de la sentencia **ATTR**.
La sentencia **ATTR** es un todo similar a la **SCREEN$**, puesto que como parámetro es necesario especificar las coordenadas de la posición de carácter concreta. En cambio, en esta ocasión obtendremos como respuesta un número, que nos indicará el estado de los atributos de la posición de pantalla (**INK, PAPER, BRIGHT** y **FALSH**). El formato general de la sentencia es:

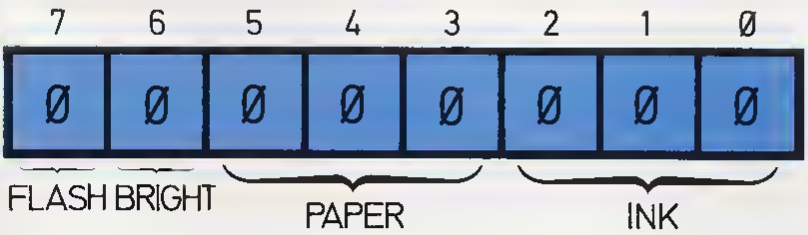
ATTR (X,Y)

Donde X e Y son las coordenadas de la posición de pantalla, y **ATTR** puede ir precedido por la setencia **PRINT**, una asignación a variable por medio de **LET**, alguna sentencia de comparación, etc., según nuestras necesidades. Sin embargo, el resultado obtenido requiere de una interpretación.

ESTRUCTURA DE LA ZONA DE ATRIBUTOS

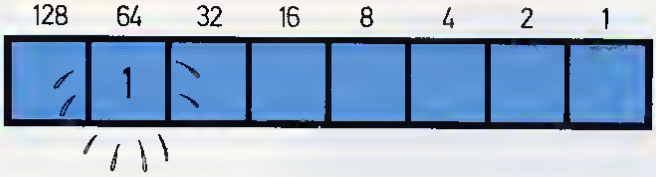
Sabemos que la zona de atributos está compuesta por 768 *bytes*, que se corresponden con cada una de las posiciones de carácter de la pantalla. En cada una de estas posiciones, se almacenan cuatro atributos: **INK, PAPER, BRIGHT** y **FLASH**.
En principio, puede parecernos difícil incluir esta

Cada byte del área de atributos contiene la información relativa a **PAPER, INK, FLASH** *y* **BRIGHT**.



¡!

Se reservan 6 144 (24 × 32 × 8) *bytes* para almacenar la configuración de puntos de la pantalla, dado que cada una de las posiciones de carácter necesita de 8 bytes (64 *bits*) para su representación.



El sexto bit otorga la cualidad de brillo extra

¡!

El valor resultante de **ATTR** es la suma de cuatro números de la siguiente forma, 128 si la posicion de carácter parpadea o 0 si no lo hace, 64 si la posición está dotada de brillo especial y 0 en caso contrario, 8 × el codigo de color del **PAPER**, más el código de color de la **INK**

*

ATTR es similar a **SCREEN$**. Da como respuesta un numero que nos indica el estado de los atributos de la posicion de pantalla (**INK**, **PAPER**, **BRIGHT** y **FLASH**). El formato general de la sentencia es

ATTR (X,Y)

información en un sólo *byte*, pero tiene su explicación. Esto se debe a que los códigos de color, tanto de **INK** como de **PAPER**, oscilan entre 0 y 7, por lo cual pueden almacenarse en *3* bits, y los atributos **BRIGHT** y **FLASH** son mucho más simples, dado que basta con conocer si están o no activos, por lo cual sólo requieren de un bit cada uno, completando así el total de 8

Por lo tanto, examinando los 8 *bits* que componen cada *byte* de la zona de atributos de izquierda a derecha, el primer *bit* corresponde al **FLASH**, el segundo al **BRIGHT**, los del 3 al 5 al **PAPER** y los del 6 al 8 a la **INK**

Con lo dicho, seremos capaces de interpretar de forma fácil la información facilitada por la sentencia **ATTR**. Para ello, debemos sólo pensar en la estructura binaria de los datos almacenados en el *byte*.

Si el valor es igual o superior a 128, ello implica que el *bit* situado en el extremo izquierdo es un 1, así pues, la posición de pantalla encuestada está parpadeando. Si el valor se encuentra, por el contrario, entre 64 y 127, la posición está dotada de brillo especial pero no de parpadeo. En general, podemos decir que el valor resultante de **ATTR** es la suma de cuatro números de la siguiente forma: 128 si la posición de carácter parpadea o 0 si no lo hace, 64 si la posición está dotada de brillo especial y 0 en caso contrario, 8 multiplicado por el código de color del **PAPER**, más el código de color de la **INK**

DECODIFICADOR DE ATTR

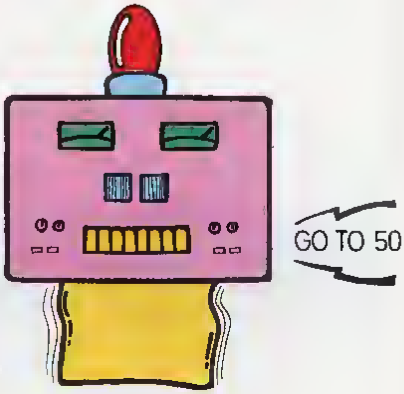
El siguiente programa, decodifica el valor del resultado suministrado por la función **ATTR**, en tinta, papel, brillo y parpadeo

```
  10 REM - DECODIFICACION ATTR - J.M.LOP
EZ MARTINEZ
  20 INPUT 'Valor ATTR:'; LINE X$: GO SU
B 100: IF X THEN  GO TO 40
  30 IF VAL X$>=0 AND VAL X$<=255 THEN
GO TO 50
  40 BEEP .25,0: GO TO 20
  50 LET X=VAL X$: PRINT ('  '+X$)(LEN X
$ TO );'=';
  60 IF X>127 THEN  LET X=X-128: PRINT '
PARPADEO ';
  70 IF X>63 THEN  LET X=X-64: PRINT 'BR
ILLO ';
  80 LET Y=INT (X/8): LET X=X-8*Y: GO SU
B 150: PRINT X$;'/';: LET X=Y: GO SUB 15
0: PRINT X$
  90 GO TO 20
 100 LET X=0
 110 FOR I=1 TO LEN X$
 120 IF X$(I)<'0' OR X$(I)>'9' THEN  LET
 X=1
 130 NEXT I
 140 RETURN
 150 RESTORE : FOR I=0 TO X: READ X$: NE
XT I: RETURN
 160 DATA 'NEGRO','AZUL','ROJO','MAGENTA
','VERDE','CIAN','AMARILLO','BLANCO'
```

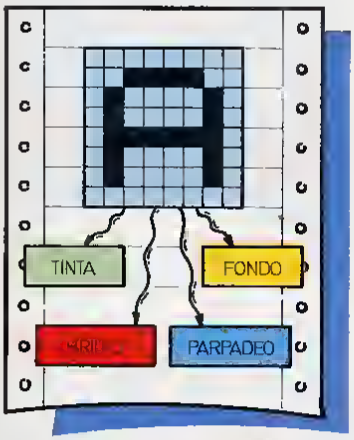
En la línea 20, proponemos una nueva técnica de **INPUT** numérico, en el cual se acepta de entrada una variable de cadena, por lo que admite cualquier tipo de caracteres. A continuación, por medio de la subrutina de la línea 100, se verifica que todos los caracteres que componen la entrada son numéricos, retornando el resultado de esta prueba en la variable **X**, cuyo valor será 1 en caso de existir caracteres no numéricos y 0 en caso contrario.

Como complemento, en la línea 30 se completa la comprobación de la coherencia del dato introducido, verificando que se encuentre entre límites aceptables (0 y 255 en ese caso), por medio

Si no se produce error, el programa continúa en la línea 50 donde se imprime el comienzo del mensaje de traducción.

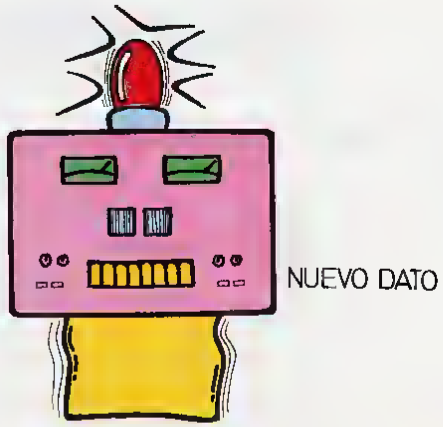


de la sentencia **VAL**. Como ya hemos visto en páginas anteriores, el uso indiscriminado de esta sentencia nos puede acarrear problemas, debido precisamente a su gran potencia, ya que el intérprete trata de obtener el valor numérico de cualquier carácter alfabético, en una variable, produciéndose el error **2 Variable not found** (Variable no encontrada).
La línea 40 canaliza la salida por error, emitiendo un tono de aviso y volviendo a pedir el dato. En caso contrario, el programa continua con la línea 50, donde se imprime el comienzo del mensaje de traducción.
En las líneas 60 y 70 se efectúa la comprobación de existencia de los atributos **FLASH** y **BRIGHT**, comparando con los valores 128 y 64, respectivamente, y restando ese número en caso afirmativo del valor de entrada.



Nuestro programa decodifica el valor suministrado por **ATTR** *en tinta, fondo, brillo y parpadeo*

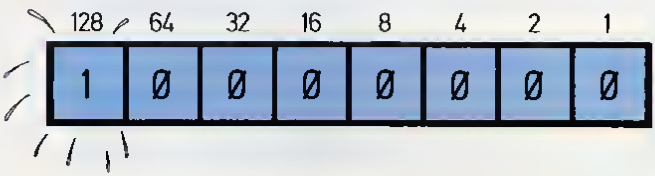
Por último, en la línea 80 se decodifica el valor del resto teniendo en cuenta su estructura binaria, en papel y tinta, ayudándose para ello de la rutina de la línea 150, que traduce los códigos de color a los nombres correspondientes a partir de la **DATA** de la línea 160.
Por si no disponemos del programa decodificador y deseamos evaluar una posición de **ATTR**, siempre podremos recurrir al siguiente método manual. Dado un valor inicial *X*:
1) Si *X* es mayor que 127, entonces restarle 128, llegando además a la conclusión de que la posición está afectada por **FLASH**. En caso contrario, dejar *X* en su estado actual, subiendo que la posición analizada carece de **FLASH**.
2) Si *X* al llegar a este paso (recordemos que ha podido ser alterado en el anterior) tiene un valor mayor que 63, deberemos restarle 64 y sabremos



La línea 40 canaliza la salida de error, emitiendo un tono de aviso y volviendo a pedir el dato

que el atributo analizado incluye brillo extra (**8BRIGHT**). De no ser este el caso, el valor de *X* permanecerá inalterado, significando esto que el carácter afectado carece de brillo extra.
3) Cualquiera que sea el valor de *X* en este momento, deberemos dividirlo por ocho y hallar su parte entera, es decir, su cociente entero sin decimales, lo cual podría conseguirse desde el BASIC mediante **PRINT INT (X/8)**. El valor obtenido indicará el codigo de color de fondo (**PAPER**).
4) Dado que en el punto anterior no se ha efectuado ninguna alteración de *X*, deberemos tomar el *X* que llego al punto tres y realizar sobre él una operación denominada módulo ocho, o lo que es lo mismo, hallar el resto de la división *X* partido por ocho. Esto puede ser efectuado de dos maneras: restandolo a *X* el producto de ocho por el valor obtenido en el paso anterior para **PAPER**, o bien ejecutando la siguiente linea BASIC **PRINT X-8*INT (X/8)**. En todo caso, el valor obtenido será el código de tinta (**INK**), con lo cual completaremos la información referente al atributo.

El bit más significativo (byte mayor o igual que 128), implica el parpadeo de la posición afectada



¡!

Examinando los 8 *bits* que componen cada *byte* de la zona de atributos de izquierda a derecha el primer *bit* corresponde al **FLASH**, el segundo al **BRIGHT** los del 3 al 5 al **PAPER** y los del 6 al 8 a la **INK**

UNA BUENA IMAGEN

ENTRE las razones que justifican el éxito obtenido por nuestro Spectrum, una de las más importantes es, sin duda, la facilidad de conexión con los periféricos básicos, como el casete o la pantalla de visualización. La consecuencia es inmediata los precios se abaratan, pues casi todos disponemos de una grabadora de mayor o menor calidad, y de no ser así, la podemos conseguir a un precio realmente económico.

Pero resulta todavía más habitual encontrar en nuestra casa un televisor, y utilizar nuestro micro con el sano propósito de efectuar la competencia a los mejores telefilmes.

En principio, no le exigiremos mucho al televisor debe estar preparado para recibir señal de UHF, y como mínimo ¡que se vea! Por supuesto, uno en color aportará resultados más espectaculares a todos nuestros juegos de acción, pero si es blanco y negro, al menos ya estaremos en condiciones de poder trabajar con nuestro Spectrum.

No quiere esto decir que sin haber efectuado la conexión entre ordenador y televisor el primero no funcione. De hecho, a partir del momento en el cual conectamos la fuente de alimentación, al microordenador, y esta a la red, nuestro Spectrum está preparado para cualquier trabajo que le sea encomendado. Pero claro, no tendremos la posibilidad de controlar el proceso a través de la pantalla, lo cual a la hora, por ejemplo, de intentar un complicado aterrizaje, puede resultar bastante peligroso para nuestro aeroplano, además de muy aburrido.

Pudiera haber quien pensara que el tamaño de la pantalla del televisor es un condicionamiento fundamental a la hora de obtener mayor resolución gráfica. En realidad, esto no es así, mayor superficie de representación no implica necesariamente mejor calidad. Y ni mucho menos, que con ello obtengamos en cada línea más de los 32 caracteres, fijados por el propio Sistema Operativo del Spectrum, grabado en la memoria R. O. M. Sencillamente, veremos todo más grande o más pequeño, en función de las dimensiones de la pantalla. La definición de la imagen estará impuesta por la calidad de los componentes utilizados en la construcción del aparato, y de los posibles mandos de ajuste que ofrezca al usuario. Pues bien, como vemos, en principio cualquier televisor puede sacarnos del atolladero, pero cuando profundicemos en las posibilidades de nuestro micro y vayamos hacia aplicaciones más profesionales, como la construcción y representación de gráficos en alta resolución, o el proceso de textos, el cual admite, por lo general, la posibilidad de presentar en la pantalla 64 caracteres en cada línea, en lugar de los 32 habituales, quizás encontremos la calidad de imagen de un televisor convencional inferior a nuestros requerimientos. Los monitores vienen a subsanar este problema.

Un televisor a color aportará resultados mas espectaculares a nuestro juegos de acción.

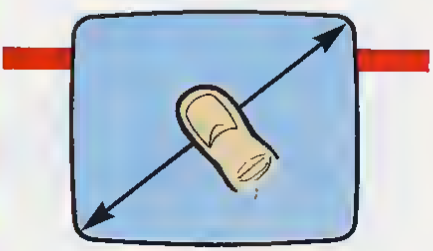
¡!

Una mayor superficie de representacion no implica necesariamente mejor calidad

*

En los Estados Unidos utilizan otro sistema de configuración de pantallas llamado NTSC

Cuando nos referimos al tamaño de la pantalla lo hacemos en pulgadas



625 LINEAS

Los principios básicos en cuanto a funcionamiento y construcción tanto de los monitores como de los aparatos de televisión, son similares. El elemento principal es el tubo de imagen, a través del cual circula un rayo catódico, barriendo la pantalla completa, a razón de 25 veces por segundo.

La superficie interior de ésta, se recubre con una serie de compuestos (normalmente a base de fósforo o sulfuros de cinc activados con plata) fotoactivos, es decir, cuando un punto recibe el impacto por parte de los electrones del haz, se excita, brillando con una determinada tonalidad, en función de la intensidad del rayo recibido. Según se trate de un aparato en blanco o negro o color, percibiremos éstos, o una escala más o menos variada de grises.

Normalmente, cuando nos referimos al tamaño de la pantalla lo hacemos en pulgadas *(inches)*, unidad inglesa equivalente a 2,54 cm. Debemos aclarar la frase: «Se trata de un televisor de 17 pulgadas». Con ello, nos referimos a que la longitud de la diagonal principal del hipotético paralelogramo formado por la pantalla, mide dicha longitud.

El ordenador funciona aunque no hayamos efectuado la conexión con el televisor, pero no podremos controlar el proceso

La definición de la imagen estará impuesta por la calidad de los componentes y los mandos de ajuste.

Hoy en día, en los televisores modernos y monitores de vídeo, la relación entre la anchura y altura de la pantalla, es de 4:3, como norma general.

Cada imagen completa de televisión (técnicamente se denomina Cuadro), no está formada por fotogramas como ocurre en el cine, sino por 625 líneas, compuesta cada una por un gran número de puntos. Estas son recorridas de arriba a abajo alternativamente por el haz electrónico, es decir, primeramente se imprimen las líneas 1, 3, 5, etc... A continuación, la 2, 4, 6..., y así, hasta el final. Luego se repite cíclicamente el proceso. A cada uno de estos semicuadros se le conoce como campo.

Debido al fenómeno de persistencia, según el cual un material fluorescente excitado necesita un cierto tiempo para volver a su estado inicial, despues de desaparecer el efecto que lo puso en situación de activo, o lo que es lo mismo, tras dejar de incidir sobre él el haz electrónico, nosotros percibimos todo el proceso como algo total. Tengamos en cuenta, que el tiempo empleado en barrer cada campo es de aproximadamente 20 milisegundos, equivalentes a una frecuencia de 50 Hz. Además, de esta manera, se consigue dismi-

¡!

La definición de la imagen esta impuesta por la calidad de los componentes utilizados en la construccion del televisor o monitor, y de los mandos de ajuste que ofrezca al usuario.

*

El elemento principal tanto en los monitores como en lo aparatos de televisión es el tubo de imagen.

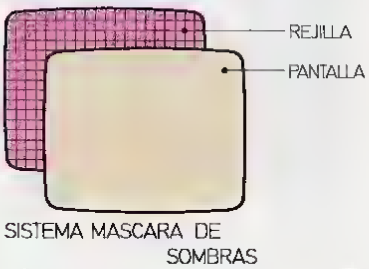
¡!

Si utilizamos nuestro micro para aplicaciones profesionales, quizás la calidad de imagen de un televisor no sea suficiente. Los monitores vienen a subsanar este problema

*

Normalmente cuando nos referimos al tamaño de la pantalla lo hacemos en pulgadas

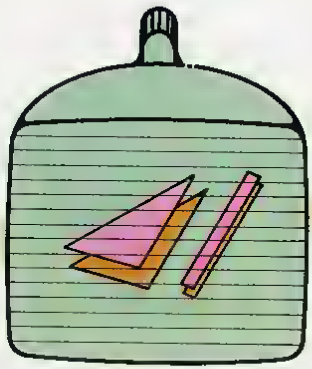
nuir el efecto estroboscópico o de parpadeo de la imagen hasta límites bajísimos.
La mayoría de los países europeos, entre ellos España, han adoptado esta norma. Pero otros, como los Estados Unidos, siguen una algo diferente, mediante la cual la pantalla queda conformada en 525 líneas, barriendo el haz cada cuadro 60 veces por segundo. Este sistema se conoce bajo las siglas N.T.S.C. (National Television System Committee, Comisión Nacional de Sistemas de Televisión).
Como vemos, el rayo de electrones que incide sobre la pantalla se mueve a altas velocidades. Para conseguirlas, el interior del tubo incorpora una serie de circuitos aceleradores, además de otros encargados de enfocarlo y dirigirlo hacia el punto exacto de la pantalla. Dentro se ha efectuado el vacío, y las tensiones en el interior son del orden de 12.000 voltios.



En el sistema de mascara de sombras se coloca próxima a la cara interior de la pantalla, una placa metalica con centenares de pequeños orificios

Una etapa elevadora de tensión es la encargada de suministrársela al tubo, de manera que éste sea capaz de crear el campo magnético necesario, para atraer hacia la pantalla las partículas electrónicas que conforman el haz.
Finalmente, una serie de circuitos tienen como misión recoger la señal exterior que recibe el equipo y coordinar las funciones de los componentes citados anteriormente. Entre ellos cabe destacar el amplificador de señal, los controladores de brillo y luminosidad de la pantalla, el oscilador de líneas (fija el número de líneas horizontales), y el oscilador de cuadro (determina el

Debido al fenómeno de persistencia percibimos una imagen como algo total



Cada imagen completa de televisión está formada por 625 líneas, compuesta cada una por un gran número de puntos.

número de imágenes a representar por segundo).
Por todo lo anteriormente considerado, debemos obrar con precaución, o lo que es mejor, dejar a un especialista las labores de manipulación de los componentes ocultos tras la carcasa del televisor o monitor. Además, debemos prestar atención especial a no golpear la pantalla, pues su rotura y subsiguiente implosión (explosión hacia adentro, debida a la presión atmosférica), puede acarrear consecuencias bastante desagradables para nuestra integridad física.

A TODO COLOR

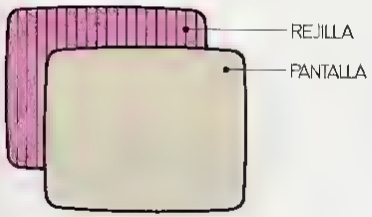
Hasta este punto, la tecnología empleada en la construcción de aparatos de televisión y monitores, ya sea en blanco y negro o color, sigue los principios básicos comentados anteriormente. Pero los receptores preparados para recibir imágenes en color sufren variaciones, las cuales involucran directamente al tubo donde se genera el haz de rayos catódicos.
Los diferentes colores se consiguen a partir de mezclas de los tres básicos: azul, rojo y verde. Por ello, al contrario de lo que sucedía en un receptor de blanco y negro, en vez de un único haz electrónico, tendremos tres independientes, emitidos a partir de tres cátodos o cañones individuales, cada uno de los cuales maneja, exclusivamente, el haz de su color correspondiente.
El recubrimiento interior de la pantalla sufre también modificaciones frente a la tecnología em-

pleada por los aparatos en blanco y negro. Existen varios sistemas de tubos de imagen de color, pero son dos los comúnmente más utilizados por los fabricantes: de máscara de sombras y Trinitrón.

El primero consiste en colocar próxima a la cara interior de la pantalla, una placa metálica (denominada máscara de sombras), sobre la cual han sido practicados del orden de medio millón de pequeños orificios, quedando distribuidos éstos regularmente. A cada uno de ellos le corresponde una serie de tres puntos (Triada) fosforescentes, sobre la superficie de la pantalla.

A través de los diminutos taladros, cruzan los tres haces electrónicos emitidos por los cañones generadores de los colores básicos. La disposición de la triada en relación con los cañones es tal que el haz rojo sólo puede incidir sobre el punto rojo, el verde sobre el verde, y el azul, solamente sobre el punto azul.

Modulando adecuadamente la intensidad de los tres haces, el ojo humano percibe toda la gama posible de colores, como combinación de los tres básicos. A la distancia normal de visión, apreciamos la pantalla, llena de imagen como un único elemento, pero si nos acercamos a ella y la observamos atentamente (mejor con la ayuda de una lupa), podemos distinguir esta estructura de puntos individuales.

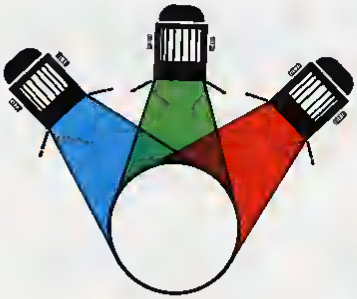


El sistema Trinitron consiste en colocar una placa metálica con cientos de rendijas próxima a la pantalla.

El segundo de los sistemas empleados en tubos de pantalla fue desarrollado por la firma SONY, con el objeto de disminuir la complejidad en equipos portátiles o de tamaño reducido. En este caso la placa metálica colocada previamente a la pantalla del tubo, está formada por varios cientos de rendijas verticales.

A través de éstas, pasan los tres haces con la información sobre los colores básicos, yendo a incidir sobre tres franjas fotoactivas, dispuestas verticalmente sobre la pared interior de la pantalla. Es decir, a cada rendija de la placa le corresponden tres franjas sobre la superficie de la pantalla. Al igual que antes, cada una de estas franjas recibe, únicamente, el impacto del chorro electrónico correspondiente al color que la activa.

Como comentábamos, el sistema da excelentes resultados en aparatos de dimensiones reducidas. Pero al aumentar el tamaño de la pantalla, las franjas de fósforo tienden a hacerse demasiado visibles, deteriorando con ello la calidad de la imagen reproducida.



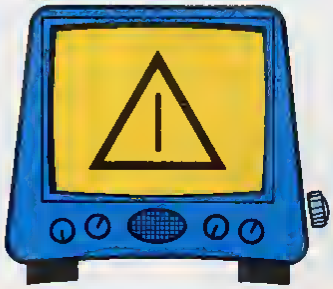
El televisor a color utiliza tres cátodos de un color básico determinado: azul, rojo y verde.

En el próximo capítulo dedicado a los terminales de pantalla, analizaremos el comportamiento del Spectrum como una emisora más de televisión, el tipo de señales que emite, dónde se producen y su compatibilidad con monitores de alta resolución gráfica.

Además, realizaremos un exhaustivo análisis de los tres sistemas comúnmente más utilizados: el televisor convencional, otro con entrada auxiliar especial para microordenadores, RF (radiofrecuencia), y finalmente, un monitor de vídeo con imagen de alta calidad.

Finalmente, comentaremos aspectos relacionados con la ergonomía (ciencia dedicada a conseguir una mayor adaptación entre el hombre y la máquina), la cual alcanza especial importancia en el diseño de vídeo terminales, pues temas como la influencia de las radiaciones, cansancio visual, etc., provocan en la actualidad polémicas entre los usuarios de estos periféricos.

Debemos tener cuidado con no golpear la pantalla del televisor o monitor, pues su rotura e implosión puede acarrear peligros a nuestra integridad física.



¡!

La superficie interior del tubo esta compuesta por materiales fotoactivos

*

Cada imagen completa de televisión está formada por 625 líneas, que son recorridas de arriba a bajo alternativamente por el haz electrónico

EL ELEFANTE

¡!

Recordemos que las letras que aparecen subrayadas en el listado corresponden a los gráficos definidos de las mismas teclas

*

Los caracteres que aparecen en el listado subrayados doblemente deben introducirse como los gráficos cambiados (con **CAPS SHIFT**) de las teclas correspondientes

Los literales subrayados en negrita **INV.** y **TRUE**, encerrados entre corchetes se corresponden a los códigos de control **INVERSE VIDEO** y **TRUE VIDEO**, obtenidos por el teclado mediante **CAPS SHIFT** + **4** y **CAPS SHIFT** + **3**, respectivamente

DENTRO de las diferentes especies de animales que pueblan el interior de la jungla de Spectrumlandia, existe un diminuto, curioso y avispado animalito: el ratoncito peluso.

Al igual que todos sus primos, este minúsculo roedor posee sus mismos hábitos y costumbres, pero difiere enormemente de aquellos en la forma de divertirse. Al ratón peluso le encanta jugar en grupo, y si hay un gran elefante por medio tanto mejor. Le chifla hacer cosquillas a los paquidermos hasta que éstos, al no poder asimilar el torrente y cúmulo de sensaciones, caen con toda su grandeza, ocasionando el jolgorio y mofa de la escuadra pelusa.

Los elefantes de la zona, debido a su gran amor propio intentan evitar el cuentro, y tratan por todos los medios de escapar a la simpática caza roedora, aún a sabiendas de que lo más seguro es que acaben muertos de risa. Sólo los paquidermos más inteligentes logran evadirse de estos pequeños roedores.

EL PROGRAMA

El programa está confeccionado para dos jugadores. Nuestro Spectrum será el árbitro de la partida, llevando en todo momento el control absoluto de la misma.

Para la adopción del listado es preciso tener en cuenta que los caracteres que aparecen con un sólo subrayado corresponden a los gráficos de la tecla afectada, y los que observan un subrayado doble, a los gráficos cambiados de dicha tecla, es decir, los obtenidos mediante **CAPS SHIFT** en combinación con el mencionado carácter, que en este caso, siempre será numérico.

Por último, los **REM** distribuidos por el programa para facilitar su lectura, se han editado en vídeo inverso, lo cual se advierte por los literales **INV.** y **TRUE** que figuran subrayados y entre corchetes.

Una vez el programa ha sido correctamente introducido y ejecutado, nuestro microordenador pedirá los nombres de los dos jugadores que se van a enfrentar. El primero dirigirá los movimientos del elefante, mientras que el segundo será el responsable de organizar el asedio del paquidermo, utilizando los cuatro ratones pelusos que el Spectrum pone a su disposición.

La hipotética jungla es representada en la pantalla por un tablero de ajedrez, en la cual los ratones pelusos aparecen en su zona superior y el elefante en la inferior.

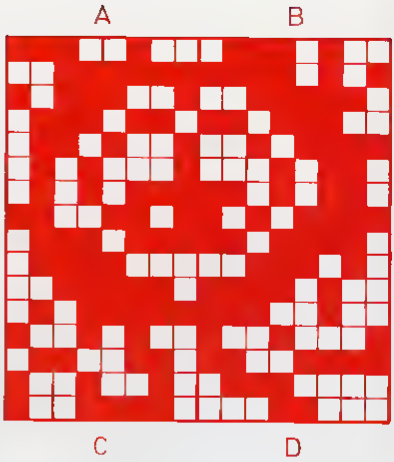
El elefante puede moverse en cualquiera de los

cuatro sentidos diagonales que permite el tablero, pulsando las siguientes teclas.

W — arriba - derecha
Q — arriba - izquierda
A — abajo - izquierda
S — abajo - derecha

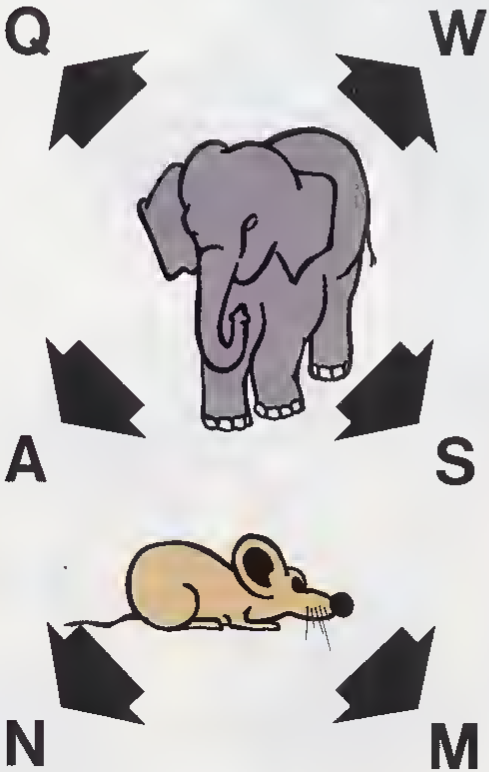
El segundo jugador deberá seguir las siguientes normas para el correcto desplazamiento de cualquiera de los componentes de su roedora escuadra pelusa.

1. Los ratones no pueden retroceder. Están condenados al avance perpetuo.
2. Cuando el Spectrum lo requiera, introducir la coordenada del ratón que se quiere mover. Ejem. A1. Para facilitar esta labor, los bordes del tablero se encuentran marcados con números y letras, de forma que se pueda obtener la coordenada mediante un simple cruce, de forma similar a la utilizada en el tradicional juego de los barcos.



La definición de los ratones se encuentra en los graficos A, B, C y D

El movimiento del elefante se consigue mediante las teclas Q, W, A, y S. El de los ratones con M y N.

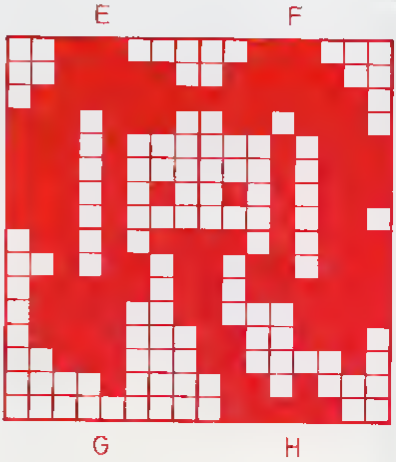


3. Pulsar cualquiera de las siguientes teclas para efectuar el movimiento deseado.

M — abajo - derecha
N — abajo - izquierda

Ya sólo nos resta desear la mejor fortuna para ambos contendientes.

Para el elefante empleamos los grafieos de E, F, G, y H



```
 10 REM **********************
 20 REM * J.M.MAYORAL SERRANO *
 30 REM **********************
 40 REM *  ELEFANTE (C) 1985  *
 50 REM **********************
 60 GO SUB 3050
 70 LET W$=G$
 80 PAPER 1: BORDER 1: INK 9
 90 CLS
100 LET JUG=0
110 POKE 23658,8
120 LET OIR=21
130 PRINT
140 PRINT
150 PRINT
160 PRINT  PAPER 6;"           JUGADOR - 1
"
170 PRINT AT 21,6; FLASH 1;"  INTRODUCE NOMBRE  "
180 INPUT "JUGADOR 1",A$
190 IF A$="" THEN  GO TO 180
200 IF LEN A$>10 THEN  GO TO 180
210 PRINT AT 5,5;A$;AT 5,17;"   (ELEFANTE)"
220 PRINT
230 PRINT  PAPER 7,AT 9,0;"           JUGADOR - 2
"
240 INPUT "JUGADOR 2",B$
250 IF B$="" THEN  GO TO 240
260 IF LEN B$>10 THEN  GO TO 240
270 PRINT AT 11,5;B$;AT 11,17;"   (RATONES)"
280 PRINT AT 21,6;"
290 LET C$=" PULSA UNA TECLA PARA EMPEZAR "
300 FOR N=1 TO LEN C$
310 IF C$(N)=" " THEN  PRINT AT 12,N;C$(N): BEEP .02
,40: GO TO 340
320 PRINT AT 18,N;C$(N)
330 BEEP .1,20
340 NEXT N
350 IF INKEY$="" THEN  BEEP .001,40: GO TO 350
360 BEEP .2,40
370 CLS
380 REM [INV.]  DIBUJO TABLERO  [TRUE]
390 REM
400 FOR F=2 TO 16 STEP 4
410 FOR C=2 TO 16 STEP 4
420 PRINT  PAPER 7;AT F,C;"  "
430 PRINT  PAPER 2;AT F,C+2;"  "
440 PRINT  PAPER 7;AT F+1,C;"  "
450 PRINT  PAPER 2,AT F+1,C+2;"  "
460 PRINT  PAPER 2;AT F+2,C;"  "
470 PRINT  PAPER 7;AT F+2,C+2;"  "
480 PRINT  PAPER 2;AT F+3,C;"  "
490 PRINT  PAPER 7;AT F+3,C+2;"  "
500 NEXT C
510 NEXT F
520 REM [INV.] NUM. TABLERO [TRUE]
530 PRINT AT 3,20; PAPER 6;"JUG-1"; PAPER 1;" (ELEF)
"
540 PRINT AT 4,21;A$
550 PRINT AT 7,20; PAPER 7;"JUG-2"; PAPER 1;" (RATN)
"
560 PRINT AT 8,21;B$
570 FOR N=1 TO 8
580 PRINT AT N*2,0;N
590 PRINT AT 0,N*2;CHR$ (64+N)
600 NEXT N
610 PRINT  INK 6;AT 1,1;"433333333333333337"
620 FOR N=2 TO 17
630 PRINT  INK 6;AT N,1;"5"
640 PRINT  INK 6,AT N,18;"5"
650 NEXT N
660 PRINT  INK 6;AT 18,1;"133333333333333332"
670 REM [INV.] POSICION FICHAS [TRUE]
680 PAPER 7: INK 9
690 FOR N=1 TO 4
700 PRINT  BRIGHT 1; INK 4;AT 2,4*N-2;"AB"
710 PRINT  BRIGHT 1; INK 4;AT 3,4*N-2;"CD"
720 NEXT N
730 PRINT  INK 1;AT 16,8;"EF"
740 PRINT  INK 1;AT 17,8;"GH"
750 FOR N=1 TO 12
760 PRINT  BRIGHT 1; INK 5; PAPER 1;AT 0,19+N;G$(N)
770 NEXT N
780 LET FE=16: LET CE=8
790 LET SWO=0: LET SW1=1
800 IF NOT SW1 THEN  GO SUB 1590
810 PRINT  PAPER 2; INK 9; FLASH 1;AT 20,7;"  MUEVE
ELEFANTE  "
820 REM [INV.] MOV. ELEFANTE [TRUE]
830 PRINT  OVER 1; PAPER 8; FLASH 1;AT 3,20,"
"
840 IF INKEY$<>"" THEN  GO TO 840
850 IF INKEY$="" THEN  GO TO 850
860 LET K$=INKEY$: BEEP .1,40
870 PRINT  OVER 1; PAPER 8, FLASH 0,AT 3,20;"
"
880 IF K$="Q" THEN  GO TO 930
890 IF K$="W" THEN  GO TO 1120
900 IF K$="A" THEN  GO TO 1300
910 IF K$="S" THEN  GO TO 1450
920 BEEP 1,-10: GO TO 830
930 REM [INV.] MOV. ELEF. ARRIBA/IZQUIERDA [TRUE]
940 IF CE-2<2 THEN  GO SUB 2320: GO TO 790
950 LET CE=CE-2
960 LET FE=FE-2
970 LET FV=FE+2
980 LET CV=CE+2
990 GO SUB 2480
1000 IF NOT SWO THEN  GO TO 1060
1010 LET FE=FE+2
1020 LET CE=CE+2
1030 LET SWO=0
1040 GO SUB 2320
1050 GO TO 790
1060 GO SUB 2510
1070 GO SUB 2880
1080 PRINT  PAPER 7; INK 7;AT FE+2,CE+2;"  "
1090 PRINT  PAPER 7; INK 7;AT FE+3,CE+2;"  "
1100 GO TO 800
1110 REM [INV.] MOV. ELEF. ARRIBA/DERECHA [TRUE]
1120 IF CE+2>16 THEN  GO SUB 2320: GO TO 790
1130 LET CE=CE+2
1140 LET FE=FE-2
1150 LET FV=FE+2
1160 LET CV=CE-2
1170 GO SUB 2480
1180 IF NOT SWO THEN  GO TO 1240
1190 LET FE=FE+2
1200 LET CE=CE-2
1210 LET SWO=0
1220 GO SUB 2320
1230 GO TO 790
1240 GO SUB 2510
1250 GO SUB 2880
1260 PRINT  PAPER 7; INK 7;AT FE+2,CE-2;"  "
1270 PRINT  PAPER 7; INK 7;AT FE+3,CE-2;"  "
1280 GO TO 800
1290 REM [INV.] MOV. ELEF. ABAJO/IZQUIERDA [TRUE]
1300 IF CE-2<2 OR FE+2>16 THEN  GO SUB 2320: GO TO 79
0
1310 LET FE=FE+2
1320 LET CE=CE-2
1330 GO SUB 2480
1340 IF NOT SWO THEN  GO TO 1400
1350 LET FE=FE-2
1360 LET CE=CE+2
1370 LET SWO=0
1380 GO SUB 2320
1390 GO TO 790
1400 GO SUB 2510
1410 PRINT  PAPER 7; INK 7;AT FE-2,CE+2;"  "
1420 PRINT  PAPER 7; INK 7;AT FE-1,CE+2;"  "
1430 GO TO 800
1440 REM [INV.] MOV. ELEF  ABAJO/DERECHA [TRUE]
1450 IF CE+2>16 OR FE+2>16 THEN  GO SUB 2320: GO TO 7
90
1460 LET CE=CE+2
1470 LET FE=FE+2
1480 GO SUB 2480
1490 IF NOT SWO THEN  GO TO 1550
1500 LET FE=FE-2
1510 LET CE=CE-2
1520 LET SWO=0
1530 GO SUB 2320
1540 GO TO 790
1550 GO SUB 2510
1560 PRINT  PAPER 7; INK 7;AT FE-2,CE-2;"  "
1570 PRINT  PAPER 7; INK 7;AT FE-1,CE-2;"  "
1580 GO TO 800
1590 REM [INV.] MOV. RATONES [TRUE]
1600 PRINT  OVER 1; PAPER 8; FLASH 1;AT 7,20;"
"
1610 PRINT AT 19,0;"ELIGE RATON A MOVER (Coordenada)"
1620 INPUT "INTRODUCE COORDENADA  ", LINE A$
1630 PRINT  PAPER 1;AT 19,0;"
"
1640 IF LEN A$>2 OR LEN A$<2 OR A$="" THEN  BEEP 1,-1
0: GO TO 1610
1650 IF CODE A$(1)>=65 AND CODE A$(1)<=72 AND CODE A$
(2)>=49 AND CODE A$(2)<=56 THEN  GO TO 1690
1660 IF CODE A$(1)>=49 AND CODE A$(1)<=56 AND CODE A$
(2)>=65 AND CODE A$(2)<=72 THEN  GO TO 1750
1670 LET A$="": GO TO 1640
1680 REM [INV.]AJUSTE COORDENADAS TP1 [TRUE]
1690 FOR N=1 TO 8
1700 LET C$=CHR$ (64+N)
1710 IF A$(1)=C$(1) THEN  LET C=2*N: LET F=2*VAL A$(2
): GO TO 1780
1720 NEXT N
1730 GO TO 1780
```

```
1740 REM [INV.]AJUSTE COORDENADAS TP2 [TRUE]
1750 LET P$=A$(2)+A$(1)
1760 LET A$=P$
1770 GO TO 1690
1780 IF ATTR (F,C)=124 THEN  GO TO 1828
1790 LET D$='   IMPOSIBLE . NO HAY RATON'
1800 GO SUB 2348
1810 GO TO 1610
1820 PRINT  PAPER 2; INK 6; FLASH 1;AT 21,5;'  EFECTU
A MOVIMIENTO  '
1830 IF INKEY$<>'' THEN  GO TO 1830
1840 IF INKEY$='' THEN  GO TO 1840
1850 LET K$=INKEY$
1860 IF K$='M' THEN  GO TO 1900
1870 IF K$='N' THEN  GO TO 2110
1880 BEEP 1,-10: GO TO 1830
1890 REM [INV.]MOV  RATON  ABAJO/DERECHA[TRUE]
1900 PRINT  PAPER 1;AT 21,0,'

1910 PRINT  OVER 1; PAPER 8; FLASH 0;AT 7,20;'

1920 LET JUG=JUG+1
1930 IF JUG=5 THEN  GO SUB 3290
1940 LET FR=F
1950 LET CR=C
1960 IF CR+2>16 THEN  GO SUB 2320: BEEP .1,40: GO TO
1605
1970 IF FR+2>16 THEN  GO SUB 2320: BEEP .1,40: GO TO
1600
1980 LET CR=CR+2
1990 LET FR=FR+2
2000 GO SUB 2570
2010 IF NOT SWO THEN  GO TO 2050
2020 LET SWO=0
2030 GO SUB 2328
2040 GO TO 1600
2050 GO SUB 2600
2060 PRINT  PAPER 7; INK 7;AT FR-2,CR-2;'  '
2070 PRINT  PAPER 7; INK 7;AT FR-1,CR-2;'  '
2080 GO SUB 2650
2090 GO TO 800
2100 REM [INV.]MOV. RATON  ABAJO/IZQUIERDA[TRUE]
2110 PRINT  PAPER 1;AT 21,0,'

2120 PRINT  OVER 1; PAPER 8; FLASH 0;AT 7,20;'

2130 LET JUG=JUG+1
2140 IF JUG=5 THEN  GO SUB 3290
2150 LET FR=F
2160 LET CR=C
2170 IF CR-2<2 THEN  GO SUB 2328: BEEP  1,40: GO TO 1
600
2180 IF FR+2>16 THEN  GO SUB 2320: BEEP .1,40: GO TO
1600
2190 LET FR=FR+2
2200 LET CR=CR-2
2210 GO SUB 2570
2220 IF NOT SWO THEN  GO TO 2260
2230 LET SWO=0
2240 GO SUB 2320
2250 GO TO 1600
2260 GO SUB 2600
2270 PRINT  PAPER 7; INK 7;AT FR-2,CR+2;'  '
2280 PRINT  PAPER 7; INK 7;AT FR-1,CR+2;'  '
2290 GO SUB 2650
2300 GO TO 800
2310 REM [INV.]  ERROR  [TRUE]
2320 PRINT  PAPER 1;AT 20,0;'

2330 LET D$='NO PUEDES HACER ESE MOVIMIENTO'
2340 FOR F=1 TO LEN D$
2350 PRINT  PAPER 1;AT 20,F;D$(F)
2360 BEEP .01,55
2370 NEXT F
2380 PRINT  OVER 1; FLASH 1;AT 20,0;'

2390 FOR M=1 TO 25
2400 BEEP .01,40
2410 NEXT M
2420 FOR F=0 TO 31
2430 PRINT  PAPER 1;AT 20,F;' '
2440 NEXT F
2450 BEEP .1,35
2460 RETURN
2470 REM [INV.] CASILLA OCUPADA [TRUE]
2480 IF ATTR (FE,CE)=56 OR ATTR (FE,CE)=63 THEN  LET
SWO=0: RETURN
2490 LET SWO=1: RETURN
2500 REM [INV.] DISP.MOV.ELEF. [TRUE]
2510 PRINT AT FE,CE; PAPER 7; INK 1;'EF'
2520 PRINT AT FE+1,CE; INK 1; PAPER 7;'GH'
2530 PRINT  PAPER 1;AT 20,0;'

2540 LET SW1=0
2550 RETURN
2560 REM [INV.] OCUP. RAT. [TRUE]
2570 IF ATTR (FR,CR )=56 OR ATTR (FR,CR)=63 THEN  LET
SWO=0: RETURN
2580 LET SWO=1: RETURN
2590 REM [INV.] MOV. RATN [TRUE]
2600 PRINT  PAPER 7; INK 4; BRIGHT 1;AT FR,CR;'AB'
2610 PRINT  PAPER 7; INK 4; BRIGHT 1;AT FR+1,CR;'CD'
2620 LET SW1=1
2630 RETURN
2640 REM [INV.] ELEF. ACORRALADO? [TRUE]
2650 IF ATTR (FE-2,CE-2)=124 AND ATTR (FE-2,CE+2)=124
 AND ATTR (FE+2,CE-2)=124 AND ATTR (FE+2,CE+2)=124 TH
EN  GO TO 2750
2660 IF CE=2 AND ATTR (FE-2,CE+2)=124 AND ATTR (FE+2,
CE+2)=124 THEN  GO TO 2750
2670 IF CE=16 AND ATTR (FE-2,CE-2)=124 AND ATTR (FE+2
,CE-2)=124 THEN  GO TO 2750
2680 IF FE=2 AND ATTR (FE+2,CE-2)=124 AND ATTR (FE+2,
CE+2)=124 THEN  GO TO 2750
2690 IF FE=16 AND ATTR (FE-2,CE-2)=124 AND ATTR (FE-2
,CE+2)=124 THEN  GO TO 2750
2700 IF FE=2 AND C-2 AND ATTR (FE+2,CE+2)=124 THEN  G
O TO 2750
2710 IF FE-2 AND CE=16 AND ATTR (FE+2,CE-2)=124 THEN
 GO TO 2750
2720 IF FE=16 AND CE=16 AND ATTR (FE-2,CE-2)=124 THEN
  GO TO 2750
2730 IF FE=16 AND CE=2 AND ATTR (FE-2,CE+2)=124 THEN
 GO TO 2750
2740 RETURN
2750 FOR N=19 TO 21
2760 PRINT  PAPER 6;AT N,0;'

2770 NEXT N
2780 LET D$='ELEFANTE ACORRALADO'
2790 FOR N=1 TO LEN D$
2800 PRINT  INK 9;AT 19,7+N;D$(N)
2810 NEXT N
2820 PRINT AT 21,1;'JUGADAS = ';JUG
2830 PRINT AT 21,16; OVER 1;'PULSA UNA TECLA'
2840 PAUSE 3
2850 IF INKEY$='' THEN  GO TO 2830
2860 BEEP  1,40: RUN 80
2870 REM [INV.]  ELEFANTE LIBRE  [TRUE]
2880 IF FE-2 AND (CE)=2 AND CE(=16) THEN  GO TO 2900
2890 RETURN
2900 PRINT  PAPER 7; INK 7;AT FV,CV;'  '
2910 PRINT  PAPER 7; INK 7;AT FV+1,CV;'  '
2920 FOR N=19 TO 21
2930 LET JUG=JUG+1
2940 FOR N=1 TO LEN G$
2950 NEXT N
2960 LET D$='EL ELEFANTE CONSIGUIO ESCAPAR!'
2970 FOR N=1 TO LEN D$
2980 PRINT AT 19,N; BRIGHT 1;D$(N)
2990 NEXT N
3000 PRINT AT 21,1;'JUGADAS = ';JUG
3010 PRINT AT 21,16; OVER 1,'PULSA UNA TECLA
3020 PAUSE 3
3030 IF INKEY$='' THEN  GO TO 2830
3040 BEEP .1,40: RUN 80
3050 REM [INV.]  GRAFICOS USUARIO  [TRUE]
3060 DATA 228,63,185,118,105,81,87,205
3070 DATA 116,245,62,220,47,22,214,175
3080 DATA 119,120,62,95,148,102,146,158
3090 DATA 222,58,244,228,145,207,112,24
3100 DATA 56,62,127,238,232,232,234,232
3110 DATA 56,124,254,110,23,23,87,22
3120 DATA 107,45,125,121,120,56,8,0
3130 DATA 215,183,191,143,206,194,104,124
3140 DATA 74,46,77,46,32,77
3150 DATA 65,89,79,82,65,76
3160 LET G$='ABCDEFGH'
3170 FOR N=1 TO LEN G$
3180 FOR L=0 TO 7
3190 READ A
3200 POKE USR G$(N)+L,A
3210 NEXT L
3220 NEXT N
3230 LET G$=''
3240 FOR N=1 TO 12
3250 READ A
3260 LET G$=G$+CHR$ A
3270 NEXT N
3280 RETURN
3290 IF CODE W$(12)<>76 OR CODE W$(5)<>32 OR CODE W$(
8)<>89 OR LEN W$<>12 THEN  GO TO 3310
3300 RETURN
3310 LET OIR=OIR-21
3320 RUN USR OIR
3330 RETURN
```

YA ESTAN EN SU QUIOSCO LOS PROGRAMAS DE RUN

HEMOS PUESTO A LA VENTA LAS 5 PRIMERAS CINTAS QUE CONTIENEN LOS PROGRAMAS PUBLICADOS EN LOS FASCICULOS 1 AL 20 AMBOS INCLUSIVE.

A PARTIR DE AHORA CADA 4 FASCICULOS APARECERA UNA CINTA CON LOS PROGRAMAS CORRESPONDIENTES.

350 ptas

ESPERAMOS DE ESTE MODO COMPLACER EL DESEO UNANIME DE NUESTROS LECTORES.